the control data processor is a general-purpose microprocessor that has an industry-standard architecture,

whereby programs for the control data processor may be developed using standard tools for the architecture.

R1.126 16. ~~3.~~ The integrated circuit set forth in claim ~~1~~ 14 wherein:

the streams of data include a serial stream and a parallel stream.

R1.126 17. ~~4.~~ The integrated circuit set forth in claim ~~1~~ 14 wherein the integrated circuit further comprises:

an aggregator that aggregates certain of the data stream processors so that the aggregated data stream processors cooperate in processing a stream of data, the aggregator including

configurable interconnections between the aggregated data stream processors;

a configurable operation coordinator that coordinates operation of the aggregated data stream processors; and

a writeable configurator that specifies the configurable interconnections and the configurable operation coordinator as required to aggregate the data stream processors.

R1.126 18. ~~5.~~ The integrated circuit set forth in claim ~~1~~ 14 wherein the integrated circuit further comprises:

a writeable configuration specifier for specifying a configuration of the data stream inputs and/or outputs; and

configuration circuitry coupled between the plurality of I/O pins and the data stream processor and responsive to the configuration specifier for configuring the inputs and/or outputs as specified by the configuration specifier,

whereby the integrated circuit may be used with a plurality of transmission protocols.

R1.126 19. ~~6.~~ The integrated circuit set forth in claim ~~1~~ 14 wherein each data stream processor further comprises:

a receive processor that operates under control of the control data processor to process the data stream received from the data stream input and/or

a transmit processor that operates under control of the control data processor to process the data stream for output to the data stream output.

Rl.126

20. ~~7.~~ The integrated circuit set forth in claim ~~6~~ 19 wherein each of the receive processor or the transmit processor further comprises:

a writeable instruction memory containing instructions; and

the receive processor or the transmit processor sequentially executes certain of the instructions to process the data stream.

Rl.126

21. ~~8.~~ The integrated circuit set forth in claim ~~6~~ 19 wherein:

the receive processor and/or the transmit processor have a plurality of processing components and are configurable to bypass one or more of the components in processing the data streams.

Rl.126

22. ~~9.~~ The integrated circuit set forth in any of claims ~~1~~ 14 through 8 wherein the integrated circuit further comprises:

Also Cancel

a context processor that responds to information received from a given data stream processor that is processing a data stream to produce information about the given data stream's context and provide the context information to the given data stream processor;

the given data stream processor using the context information to process the data stream.

Rl.126

23. ~~10.~~ The integrated circuit set forth in any one of claims ~~1~~ 14 through ~~8~~ 21 wherein:

Also Cancel from Dependency

a stream of data contains control data and payload;

a received stream of data is processed in a receiving data stream processor to extract the control data and the payload and a transmitted stream of data is processed in a transmitting data stream processor to add control data to the payload; and

the integrated circuit further comprises

a buffer manager coupled to the data stream processors that provides addresses of buffers for storing payload and responds to a write operation with a buffer address to write payload to the addressed buffer and to a read operation with a buffer address to read payload from the addressed buffer; and

a queue manager coupled to the data stream processors that manages queues of descriptors of payload, each descriptor including at least a buffer address, the queue manager responding to an enqueue command by enqueuing a descriptor provided with the command to a queue specified in the command and responding to a dequeue command by dequeuing a descriptor from the queue specified in the command,